Normalized Wasserstein for Mixture Distributions with Applications in Adversarial Learning and Domain Adaptation

Yogesh Balaji
Department of Computer Science
University of Maryland
yogesh@cs.umd.edu

Rama Chellappa
UMIACS
University of Maryland
rama@umiacs.umd.edu

Soheil Feizi
Department of Computer Science
University of Maryland
sfeizi@cs.umd.edu

Abstract

Understanding proper distance measures between distributions is at the core of several learning tasks such as generative models, domain adaptation, clustering, etc. In this work, we focus on mixture distributions that arise naturally in several application domains where the data contains different sub-populations. For mixture distributions, established distance measures such as the Wasserstein distance do not take into account imbalanced mixture proportions. Thus, even if two mixture distributions have identical mixture components but different mixture proportions, the Wasserstein distance between them will be large. This often leads to undesired results in distance-based learning methods for mixture distributions. In this paper, we resolve this issue by introducing the Normalized Wasserstein measure. The key idea is to introduce mixture proportions as optimization variables, effectively normalizing mixture proportions in the Wasserstein formulation. Using the proposed normalized Wasserstein measure leads to significant performance gains for mixture distributions with imbalanced mixture proportions compared to the vanilla Wasserstein distance. We demonstrate the effectiveness of the proposed measure in GANs, domain adaptation and adversarial clustering in several benchmark datasets.

1. Introduction

Quantifying distances between probability distributions is a fundamental problem in machine learning and statistics with several applications in generative models, domain adaptation, clustering, etc. Popular probability distance measures include *optimal transport* measures such as the Wasserstein distance [22] and *divergence* measures such as the Kullback-Leibler (KL) divergence [4].

Classical distance measures, however, can lead to some issues for mixture distributions. A *mixture distribution* is the probability distribution of a random variable X where $X = X_i$ with probability π_i for $1 \leq i \leq k$. k is the number of mixture components and $\pi = [\pi_1, ..., \pi_k]^T$ is the vector of *mixture (or mode) proportions*. The probability distribution of each X_i is referred to as a *mixture component* (or, a mode). Mixture distributions arise naturally in different applications where the data contains two or more sub-populations. For example, image datasets with different labels can be viewed as a mixture (or, multi-modal) distribution where samples with the same label characterize a specific mixture component.

If two mixture distributions have exactly same mixture components (i.e. same X_i's) with different mixture proportions (i.e. different π's), classical distance measures between the two will be large. This can lead to undesired results in several distance-based machine learning methods. To illustrate this issue, consider the *Wasserstein distance* between two distributions $\mathbb{P}_X$ and $\mathbb{P}_Y$, defined as [22]

$$W(\mathbb{P}_X, \mathbb{P}_Y) := \min_{\mathbb{P}_{X,Y}} \mathbb{E}\left[\|X - Y\|\right], \quad (1)$$
$$\text{marginal}_X(\mathbb{P}_{X,Y}) = \mathbb{P}_X, \ \text{marginal}_Y(\mathbb{P}_{X,Y}) = \mathbb{P}_Y$$

where $\mathbb{P}_{X,Y}$ is the joint distribution (or coupling) whose marginal distributions are equal to $\mathbb{P}_X$ and $\mathbb{P}_Y$. When no confusion arises and to simplify notation, in some equations, we use $W(X, Y)$ notation instead of $W(\mathbb{P}_X, \mathbb{P}_Y)$.

The Wasserstein distance optimization is over all joint distributions (couplings) $\mathbb{P}_{X,Y}$ whose marginal distributions *match* exactly with input distributions $\mathbb{P}_X$ and $\mathbb{P}_Y$. This requirement can cause issues when $\mathbb{P}_X$ and $\mathbb{P}_Y$ are mixture distributions with different mixture proportions. In this case, due to the marginal constraints, samples belonging to very different mixture components will have to be coupled together in $\mathbb{P}_{X,Y}$ (e.g. Figure 1(a)). Thus, using this distance measure can then lead to undesirable outcomes in problems such as domain adaptation. This motivates the need for developing a new distance measure to take into account mode imbalances in mixture distributions.

In this paper, we propose a new distance measure that resolves the issue of imbalanced mixture proportions for

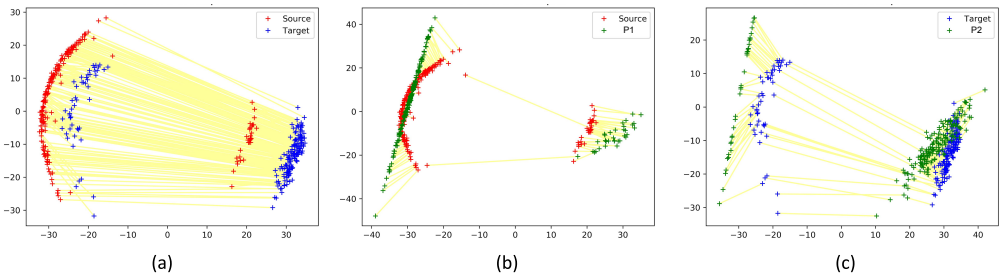

Figure 1. An illustration of the effectiveness of the proposed Normalized Wasserstein measure in domain adaptation. The source domain (shown in red) and the target domain (shown in blue) have two modes with different mode proportions. (a) The couplings computed by estimating Wasserstein distance between source and target distributions (shown in yellow lines) match several samples from incorrect and distant mode components. (b,c) Our proposed normalized Wasserstein measure (3) constructs intermediate mixture distributions $\mathbb{P}_1$ and $\mathbb{P}_2$ (shown in green) with similar mixture components to source and target distributions, respectively, but with optimized mixture proportions. This significantly reduces the number of couplings between samples from incorrect modes and leads to 42% decrease in target loss in domain adaptation compared to the baseline.

multi-modal distributions. Our developments focus on a class of optimal transport measures, namely the Wasserstein distance Eq (1). However, our ideas can be extended naturally to other distance measures (eg. adversarial distances [6]) as well.

Let $\mathbf{G}$ be an array of generator functions with k components defined as $\mathbf{G} := [\mathbf{G}_1, ..., \mathbf{G}_k]$. Let $\mathbb{P}_{\mathbf{G},\pi}$ be a mixture probability distribution for a random variable X where $X = \mathbf{G}_i(Z)$ with probability π_i for $1 \leq i \leq k$. Throughout the paper, we assume that Z has a normal distribution.

By relaxing the marginal constraints of the classical Wasserstein distance (1), we introduce the *Normalized Wasserstein measure* (*NW measure*) as follows:

$$W_N(\mathbb{P}_X, \mathbb{P}_Y) := \min_{\mathbf{G},\pi^{(1)},\pi^{(2)}} W(\mathbb{P}_X, \mathbb{P}_{\mathbf{G},\pi^{(1)}}) + W(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G},\pi^{(2)}}).$$

There are two key ideas in this definition that help resolve mode imbalance issues for mixture distributions. First, instead of directly measuring the Wasserstein distance between $\mathbb{P}_X$ and $\mathbb{P}_Y$, we construct two intermediate (and potentially mixture) distributions, namely $\mathbb{P}_{\mathbf{G},\pi^{(1)}}$ and $\mathbb{P}_{\mathbf{G},\pi^{(2)}}$. These two distributions have the same mixture components (i.e. same $\mathbf{G}$) but can have different mixture proportions (i.e. $\pi^{(1)}$ and $\pi^{(2)}$ can be different). Second, mixture proportions, $\pi^{(1)}$ and $\pi^{(2)}$, are considered as optimization variables. This effectively *normalizes* mixture proportions before Wasserstein distance computations. See an example in Figure 1 (b, c) for a visualization of $\mathbb{P}_{\mathbf{G},\pi^{(1)}}$ and $\mathbb{P}_{\mathbf{G},\pi^{(2)}}$, and the re-normalization step.

In this paper, we show the effectiveness of the proposed Normalized Wasserstein measure in three application domains. In each case, the performance of our proposed method significantly improves against baselines when input datasets are mixture distributions with imbalanced mixture proportions. Below, we briefly highlight these results:

Domain Adaptation: In Section 4, we formulate the problem of domain adaptation as minimizing the normalized Wasserstein measure between source and target feature distributions. On classification tasks with imbalanced datasets, our method significantly outperforms baselines (e.g. $\sim 20\%$ gain in synthetic to real adaptation on VISDA-3 dataset).

GANs: In Section 5, we use the normalized Wasserstein measure in GAN's formulation to train mixture models with varying mode proportions. We show that such a generative model can help capture rare modes, decrease the complexity of the generator, and re-normalize an imbalanced dataset.

Adversarial Clustering: In Section 6, we formulate the clustering problem as an adversarial learning task using Normalized Wasserstein measure.

2. Normalized Wasserstein Measure

In this section, we introduce the *normalized Wasserstein* measure and discuss its properties. Recall that $\mathbf{G}$ is an array of generator functions defined as $\mathbf{G} := [\mathbf{G}_1, ..., \mathbf{G}_k]$ where $\mathbf{G}_i : \mathbb{R}^r \to \mathbb{R}^d$. Let $\mathcal{G}$ be the set of all possible $\mathbf{G}$ function arrays. Let π be a discrete probability mass function with k elements, i.e. $\pi = [\pi_1, \pi_2, \cdots, \pi_k]$ where $\pi_i \geq 0$ and $\sum_i \pi_i = 1$. Let Π be the set of all possible π's.

Let $\mathbb{P}_{\mathbf{G},\pi}$ be a mixture distribution, i.e. it is the probability distribution of a random variable X such that $X = \mathbf{G}_i(Z)$ with probability π_i for $1 \leq i \leq k$. We assume that Z has a normal density, i.e. $Z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We refer to $\mathbf{G}$ and π as mixture components and proportions, respectively. The set of all such mixture distributions is defined as:

$$\mathcal{P}_{\mathbf{G},k} := \{\mathbb{P}_{\mathbf{G},\pi} : \mathbf{G} \in \mathcal{G}, \pi \in \Pi\} \tag{2}$$

where k is the number of mixture components. Given two distributions $\mathbb{P}_X$ and $\mathbb{P}_Y$ belonging to the family of mixture distributions $\mathcal{P}_{\mathbf{G},k}$, we are interested in defining a distance

measure agnostic to differences in mode proportions, but sensitive to shifts in mode components, i.e., the distance function should have high values only when mode components of $\mathbb{P}_X$ and $\mathbb{P}_Y$ differ. If $\mathbb{P}_X$ and $\mathbb{P}_Y$ have the same mode components but differ only in mode proportions, the distance should be low.

The main idea is to introduce mixture proportions as optimization variables in the Wasserstein distance formulation (1). This leads to the following distance measure which we refer to as the *Normalized Wasserstein measure* (NW measure), $W_N(\mathbb{P}_X, \mathbb{P}_Y)$, defined as:

$$\min_{\mathbf{G},\pi^{(1)},\pi^{(2)}} \; W(\mathbb{P}_X, \mathbb{P}_{\mathbf{G},\pi^{(1)}}) + W(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G},\pi^{(2)}}) \qquad (3)$$

$$\sum_{j=1}^{k} \pi_j^{(i)} = 1 \quad i = 1, 2,$$

$$\pi_j^{(i)} \geq 0 \qquad 1 \leq j \leq k, \;\; i = 1, 2.$$

Since the normalized Wasserstein's optimization (3) includes mixture proportions $\pi^{(1)}$ and $\pi^{(2)}$ as optimization variables, if two mixture distributions have similar mixture components with different mixture proportions (i.e. $\mathbb{P}_X = \mathbb{P}_{\mathbf{G},\pi^{(1)}}$ and $\mathbb{P}_Y = \mathbb{P}_{\mathbf{G},\pi^{(2)}}$), although the Wasserstein distance between the two can be large, the introduced normalized Wasserstein measure between the two will be zero. Note that W_N is defined with respect to a set of generator functions $\mathbf{G} = [\mathbf{G}_1, ..., \mathbf{G}_k]$. However, to simplify the notation, we make this dependency implicit. We would like to point our that our proposed NW measure is a *semi-distance* measure (and not a distance) since it does not satisfy all properties of a distance measure. Please refer to Supplementary material for more details.

To compute the NW measure, we use an alternating gradient descent approach similar to the dual computation of the Wasserstein distance [1]. Moreover, we impose the π constraints using a soft-max function. Please refer to Section 3 of Supplementary material for more details.

To illustrate how NW measure is agnostic to mode imbalances between distributions , consider an unsupervised *domain adaptation* problem with MNIST-2 (i.e. a dataset with two classes: digits 1 and 2 from MNIST) as the source dataset, and noisy MNIST-2 (i.e. a noisy version of it) as the target dataset (details of this example is presented in Section 4.2). The source dataset has $4/5$ digits one and $1/5$ digits two, while the target dataset has $1/5$ noisy digits one and $4/5$ noisy digits two. The couplings produced by estimating the Wasserstein distance between the two distributions is shown in yellow lines in Figure 1-a. We observe that there are many couplings between samples from incorrect mixture components. The normalized Wasserstein measure, on the other hand, constructs intermediate mode-normalized distributions $\mathbb{P}_1$ and $\mathbb{P}_2$, which get coupled to the correct modes of source and target distributions, respectively (see panels (b) and (c) in Figure 1)).

3. Theoretical Results

For NW measure to work effectively, the number of modes k in NW formulation (Eq. (3)) must be chosen appropriately. For instance, given two mixture distributions with k components each, Normalized Wasserstein measure with $2k$ modes would always give 0 value. In this section, we provide some theoretical conditions under which the number of modes can be estimated accurately. We begin by making the following assumptions for two mixture distributions X and Y whose NW distance we wish to compute:

- (A1) If mode i in distribution X and mode j in distribution Y belong to the same mixture component, then their Wasserstein distance is $\leq \epsilon$ i.e., if X_i and Y_j correspond to the same component, $W(\mathbb{P}_{X_i}, \mathbb{P}_{Y_j}) < \epsilon$.

- (A2) The minimum Wasserstein distance between any two modes of one mixture distribution is at least δ i.e., $W(\mathbb{P}_{X_i}, \mathbb{P}_{X_j}) > \delta$ and $W(\mathbb{P}_{Y_i}, \mathbb{P}_{Y_j}) > \delta \; \forall i \neq j$. Also, non-overlapping modes between X and Y are separated by δ i.e., for non-overlapping modes X_i and Y_j, $W(\mathbb{P}_{X_i}, \mathbb{P}_{Y_j}) > \delta$. This ensures that modes are well-separated.

- (A3) We assume that each mode X_i and Y_i have density at least η i.e., $\mathbb{P}_{X_i} \geq \eta \quad \forall i$, $\mathbb{P}_{Y_i} \geq \eta \quad \forall i$. This ensures that every mode proportion is at least η.

- (A4) Each generator $\mathbf{G}_i$ is powerful enough to capture exactly one mode of distribution $\mathbb{P}_X$ or $\mathbb{P}_Y$.

Theorem 1 *Let $\mathbb{P}_X$ and $\mathbb{P}_Y$ be two mixture distributions satisfying (A1)-(A4) with n_1 and n_2 mixture components, respectively, where r of them are overlapping. Let $k^* = n_1 + n_2 - r$. Then, k^* is smallest k for which $NW(k)$ is small ($O(\epsilon)$) and $NW(k) - NW(k-1)$ is relatively large (in the $O(\delta\eta)$)*

The proof is presented in the Section 1 of supplementary material. All assumptions made are reasonable: (A1)-(A3) enforces that non-overlapping modes in mixture distributions are separated, and overlapping modes are close in Wasserstein distance. To enforce (A4), we need to prevent multi-mode generation in one mode of $\mathbf{G}$. This can be satisfied by using the regularizer in Eq. (11). Note that in the above theorem, k^* is the optimal k that should be used in the Normalized Wasserstein formulation. The theorem presents a way to estimate k^*. Please refer to Section 7 for experimental results. In many applications like domain adaption, however, the number of components k is known beforehand, and this step can be skipped.

4. Normalized Wasserstein in Domain Adaptation

In this section, we demonstrate the effectiveness of the NW measure in Unsupervised Domain Adaptation (UDA) both for supervised (e.g. classification) and unsupervised (e.g. denoising) tasks. Note that the term *unsupervised* in UDA means that the label information in the target domain is unknown while *unsupervised* tasks mean that the label information in the source domain is unknown.

First, we consider domain adaptation for a classification task. Let (X_s, Y_s) represent the source domain while (X_t, Y_t) denote the target domain. Since we deal with the classification setup, we have $Y_s, Y_t \in \{1, 2, ..., k\}$. A common formulation for the domain adaptation problem is to transform X_s and X_t to a feature space where the *distance* between the source and target feature distributions is sufficiently small, while a good classifier can be computed for the source domain in that space [6]. In this case, one solves the following optimization:

$$\min_{f \in \mathcal{F}} \; L_{cl}\left(f(X_s), Y_s\right) + \lambda \, \text{dist}\left(f(X_s), f(X_t)\right) \tag{4}$$

where λ is an adaptation parameter and L_{cl} is the empirical classification loss function (e.g. the cross-entropy loss). The distance function between distributions can be adversarial distances [6, 21], the Wasserstein distance [20], or MMD-based distances [14, 15].

When X_s and X_t are mixture distributions (which is often the case as each label corresponds to one mixture component) with different mixture proportions, the use of these classical distance measures can lead to the computation of inappropriate transformation and classification functions. In this case, we propose to use the NW measure as the distance function. Computing the NW measure requires training mixture components $\mathbf{G}$ and mode proportions $\pi^{(1)}, \pi^{(2)}$. To simplify the computation, we make use of the fact that labels for the source domain (i.e. Y_s) are known, thus source mixture components can be identified using these labels. Using this information, we can avoid the need for computing $\mathbf{G}$ directly and use the conditional source feature distributions as a proxy for the mixture components as follows:

$$\begin{aligned} &\mathbf{G}_i(Z) \overset{\text{dist}}{=} f(X_s^{(i)}), \\ &X_s^{(i)} = \{X_s | Y_s = i\}, \quad \forall 1 \leq i \leq k, \end{aligned} \tag{5}$$

where $\overset{\text{dist}}{=}$ means matching distributions. Using (5), the formulation for domain adaptation can be written as

$$\min_{f \in \mathcal{F}} \; \min_{\pi} \; L_{cl}\left(X_s, Y_s\right) + \lambda W \left(\sum_i \pi^{(i)} f(X_s^{(i)}), f(X_t) \right). \tag{6}$$

The above formulation can be seen as a version of instance weighting as source samples in $X_s^{(i)}$ are weighted by π_i. Instance weighting mechanisms have been well studied for domain adaptation [23, 24]. However, different from these approaches, we train the mode proportion vector π in an end-to-end fashion using neural networks and integrate the instance weighting in a Wasserstein optimization. Of more relevance to our work is the method proposed in [3], where the instance weighting is trained end-to-end in a neural network. However, in [3], instance weights are maximized with respect to the Wasserstein loss, while we show that the mixture proportions need to minimized to normalize mode mismatches. Moreover, our NW measure formulation can handle the case when mode assignments for source embeddings are unknown (as we discuss in Section 4.2). This case cannot be handled by the approach presented in [3].

For unsupervised tasks when mode assignments for source samples are unknown, we cannot use the simplified formulation of (5). In that case, we use a domain adaptation method solving the following optimization:

$$\min_{f \in \mathcal{F}} \; L_{unsup}\left(X_s\right) + \lambda W_N \left(f(X_s), f(X_t)\right), \tag{7}$$

where $L_{unsup}(X_s)$ is the loss corresponding to the desired *unsupervised* learning task on the source domain data.

4.1. UDA for supervised tasks

4.1.1 MNIST → MNIST-M

In the first set of experiments[1], we consider adaptation between MNIST→ MNIST-M datasets. We consider three settings with imbalanced class proportions in source and target datasets: 3 modes, 5 modes, and 10 modes. More details can be found in Table 3 of Supplementary material.

We use the same architecture as [6] for feature network and discriminator. We compare our method with the following approaches: (1) Source-only which is a baseline model trained only on source domain with no domain adaptation performed, (2) DANN [6], a method where adversarial distance between source and target distibutions is minimized, and (3) Wasserstein [20] where Wasserstein distance between source and target distributions is minimized. Table 1 summarizes our results of this experiment. We observe that performing domain adaptation using adversarial distance and Wasserstein distance leads to decrease in performance compared to the baseline model. This is an outcome of not accounting for mode imbalances, thus resulting in negative transfer, i.e., samples belonging to incorrect classes are coupled and getting pushed to be close in the embedding space. Our proposed NW measure, however, accounts for mode imbalances and leads to a significant boost in performance in all three settings.

[1]Code available at https://github.com/yogeshbalaji/Normalized-Wasserstein

Table 1. Mean classification accuracies (in %) averaged over 5 runs on imbalanced MNIST→MNIST-M adaptation

Method	3 modes	5 modes	10 modes
Source only	66.63	67.44	63.17
DANN	62.34	57.56	59.31
Wasserstein	61.75	60.56	58.22
NW	**75.06**	**76.16**	**68.57**

4.1.2 VISDA

In the experiment of Section 4.1.1 on digits dataset, models have been trained from scratch. However, a common practice used in domain adaptation is to transfer knowledge from a pretrained network (eg. models trained on ImageNet) and fine-tune on the desired task. To evaluate the performance of our approach in such settings, we consider adaptation on the VISDA dataset [18]; a recently proposed benchmark for adapting from synthetic to real images.

We consider a subset of the entire VISDA dataset containing the following three classes: *aeroplane*, *horse* and *truck*. The source domain contains $(0.55, 0.33, 0.12)$ fraction of samples per class, while that of the target domain is $(0.12, 0.33, 0.55)$. We use a Resnet-18 model pre-trained on ImageNet as our feature network. As shown in Table 2, our approach significantly improves the domain adaptation performance over the baseline and other compared methods.

Table 2. Mean classification accuracies (in %) averaged over 5 runs on synthetic to real adaptation on VISDA dataset (3 classes)

Method	Accuracy (in %)
Source only	53.19
DANN	68.06
Wasserstein	64.84
Normalized Wasserstein	**73.23**

4.1.3 Mode balanced datasets

The previous two experiments demonstrated the effectiveness of our method when datasets are imbalanced. In this section, we study the case where source and target domains have mode-balanced datasets – the standard setting considered in the most domain adaptation methods. We perform experiment on MNIST→MNIST-M adaptation using the entire dataset. Table 3 reports the results obtained. We observe that our approach performs on-par with the standard wasserstein distance minimization.

Table 3. Domain adaptation on mode-balanced datasets: MNIST→MNIST-M. Average classification accuracies averaged over 5 runs are reported

Method	Classification accuracy (in %)
Source only	60.22
DANN	85.24
Wasserstein	83.47
Normalized Wasserstein	84.16

4.2. UDA for unsupervised tasks

For unsupervised tasks on mixture datasets, we use the formulation of Eq (7) to perform domain adaptation. To empirically validate this formulation, we consider the image denoising problem. The source domain consists of digits $\{1, 2\}$ from MNIST dataset as shown in Fig 2(a). Note that the color of digit 2 is inverted. The target domain is a noisy version of the source, i.e. source images are perturbed with random $i.i.d$ Gaussian noise $\mathcal{N}(0.4, 0.7)$ to obtain target images. Our dataset contains $5,000$ samples of digit 1 and $1,000$ samples of digit 2 in the source domain, and $1,000$ samples of noisy digit 1 and $5,000$ samples of noisy digit 2 in the target. The task is to perform image denoising by dimensionaly reduction, i.e., given a target domain image, we need to reconstruct the corresponding clean image that looks like the source. We assume that no (source, target) correspondence is available in the dataset.

To perform denoising when the (source, target) correspondence is unavailable, a natural choice would be to minimize the reconstruction loss in source while minimizing the distance between source and target embedding distributions. We use the NW measure as our choice of distance measure. This results in the following optimization:

$$\min_{f,g} \ \mathbb{E}_{\mathbf{x}\sim X_s}\|g(f(\mathbf{x})) - \mathbf{x}\|_2^2 + \lambda W_N\left(f(X_s), f(X_t)\right)$$

where $f(.)$ is the encoder and $g(.)$ is the decoder.

As our baseline, we consider a model trained only on source using a quadratic reconstruction loss. Fig 2(b) shows source and target embeddings produced by this baseline. In this case, the source and the target embeddings are distant from each other. However, as shown in Fig 2(c), using the NW formulation, the distributions of source and target embeddings match closely (with estimated mode proportions) . We measure the L_2 reconstruction loss of the target domain, $err_{recons,tgt} = \mathbb{E}_{\mathbf{x}\sim X_t}\|g(f(\mathbf{x})) - \mathbf{x}\|_2^2$, as a quantitative evaluation measure. This value for different approaches is shown in Table 4. We observe that our method outperforms the compared approaches.

Table 4. $err_{recons,tgt}$ for an image denoising task

Method	$err_{recons,tgt}$
Source only	0.31
Wasserstein	0.52
Normalized Wasserstein	0.18
Training on target (Oracle)	0.08

5. Normalized Wasserstein GAN

Learning a probability model from data is a fundamental problem in statistics and machine learning. Building on the success of deep learning, a recent approach to this problem is using Generative Adversarial Networks (GANs) [8].

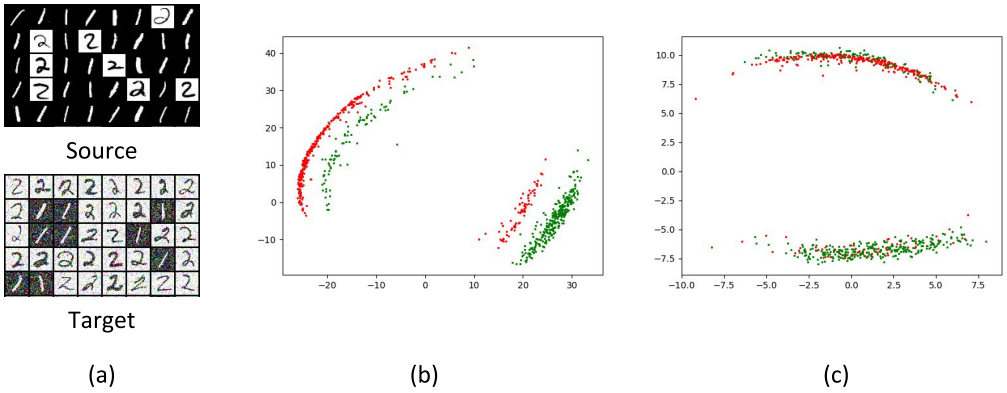

Figure 2. Domain adaptation for image denoising. (a) Samples from source and target domains. (b) Source and target embeddings learnt by the baseline model. (c) Source and target embeddings learnt by minimizing the proposed NW measure. In (b) and (c), red and green points indicate source and target samples, respectively.

GANs view this problem as a *game* between a *generator* whose goal is to generate fake samples that are close to the real data training samples, and a *discriminator* whose goal is to distinguish between the real and fake samples.

Most GAN frameworks can be viewed as methods that *minimize* a distance between the observed probability distribution, $\mathbb{P}_X$, and the generative probability distribution, $\mathbb{P}_Y$, where $Y = \mathbf{G}(Z)$. $\mathbf{G}$ is referred to as the generator function. In several GAN formulations, the *distance* between $\mathbb{P}_X$ and $\mathbb{P}_Y$ is formulated as another optimization which characterizes the discriminator. Several GAN architectures have been proposed in the last couple of years. A summarized list includes GANs based on **optimal transport** measures (e.g. Wasserstein GAN+Weight Clipping [1], WGAN+Gradient Penalty [9]), GANs based on **divergence** measures (e.g. the original GAN's formulation [8], DCGAN [19], f-GAN [17]), GANs based on **moment-matching** (e.g. MMD-GAN [5, 11]), and other formulations (e.g. Least-Squares GAN [16], BigGAN [2], etc.)

If the observed distribution $\mathbb{P}_X$ is a mixture one, the proposed normalized Wasserstein measure (3) can be used to compute a generative model. Instead of estimating a single generator $\mathbf{G}$ as done in standard GANs, we estimate a mixture distribution $\mathbb{P}_{\mathbf{G},\pi}$ using the proposed NW measure. We refer to this GAN as the *Normalized Wasserstein GAN* (or NWGAN) formulated as the following optimization:

$$\min_{\mathbf{G},\pi} \; W_N(\mathbb{P}_X, \mathbb{P}_{\mathbf{G},\pi}). \tag{8}$$

In this case, the NW distance simplifies as

$$\begin{aligned}
&\min_{\mathbf{G},\pi} W_N(\mathbb{P}_X, \mathbb{P}_{\mathbf{G},\pi}) \\
&= \min_{\mathbf{G},\pi} \min_{\mathbf{G}',\pi^{(1)},\pi^{(2)}} W(\mathbb{P}_X, \mathbb{P}_{\mathbf{G}',\pi^{(1)}}) + W(\mathbb{P}_{\mathbf{G},\pi}, \mathbb{P}_{\mathbf{G}',\pi^{(2)}}) \\
&= \min_{\mathbf{G},\pi} W(\mathbb{P}_X, \mathbb{P}_{\mathbf{G},\pi}).
\end{aligned} \tag{9}$$

There are couple of differences between the proposed NWGAN and the existing GAN architecures. The generator in the proposed NWGAN is a mixture of k models, each producing π_i fraction of generated samples. We select k a priori based on the application domain while π is computed within the NW distance optimization. Modeling the generator as a mixture of k neural networks has also been investigated in some recent works [10, 7]. However, these methods assume that the mixture proportions π are known beforehand, and are held fixed during the training. In contrast, our approach is more general as the mixture proportions are also optimized. Estimating mode proportions have several important advantages: (1) we can estimate rare modes, (2) an imbalanced dataset can be re-normalized, (3) by allowing each $\mathbf{G}_i$ to focus only on one part of the distribution, the quality of the generative model can be improved while the complexity of the generator can be reduced. In the following, we highlight these properties of NWGAN on different datasets.

5.1. Mixture of Gaussians

First, we present the results of training the NWGAN on a two dimensional mixture of Gaussians. The input data is a mixure of 9 Gaussians, each centered at a vertex of a 3×3 grid as shown in Figure 3. The mean and the covariance matrix for each mode are randomly chosen. The mode proportion for mode i is chosen as $\pi_i = \frac{i}{45}$ for $1 \leq i \leq 9$.

Generations produced by NWGAN using $k = 9$ affine generator models on this dataset is shown in Figure 3. We also compare our method with WGAN [1] and MGAN [10]. Since MGAN does not optimize over π, we assume uniform mode proportions ($\pi_i = 1/9$ for all i). To train WGAN, a non-linear generator function is used since a single affine function cannot model a mixture of Gaussian distribution.

To evaluate the generative models, we report the following quantitative scores: (1) the average mean error which

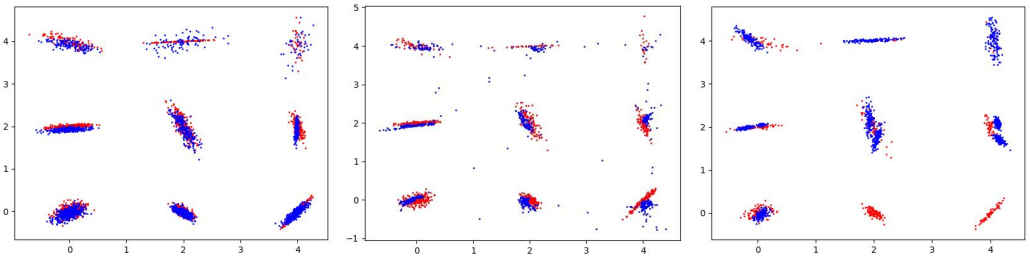

Figure 3. Mixture of Gaussian experiments. In all figures, red points indicate samples from the real data distribution while blue points indicate samples from the generated distribution. NWGAN is able to capture rare modes in the data and produces a significantly better generative model than other methods.

is the mean-squared error (MSE) between the mean vectors of real and generated samples per mode averaged over all modes, (2) the average covariance error which is the MSE between the covariance matrices of real and generated samples per mode averaged over all modes, and (3) the π estimation error which is the normalized MSE between the π vector of real and generated samples. Note that computing these metrics require mode assignments for generated samples. This is done based on the closeness of generative samples to the ground-truth means.

We report these error terms for different GANs in Table 5. We observe that the proposed NWGAN achieves best scores compared to the other two approaches. Also, from Figure 3, we observe that the generative model trained by MGAN misses some of the rare modes in the data. This is because of the error induced by assuming fixed mixture proportions when the ground-truth π is non-uniform. Since the proposed NWGAN estimates π in the optimization, even rare modes in the data are not missed. This shows the importance of estimating mixture proportions specially when the input dataset has imbalanced modes.

Table 5. Quantitative Evaluation on Mixture of Gaussians

Method	Avg. μ error	Avg. Σ error	π error
WGAN	0.007	0.0003	0.0036
MGAN	0.007	0.0002	0.7157
NWGAN	**0.002**	**0.0001**	**0.0001**

5.2. A Mixture of CIFAR-10 and CelebA

One application of learning mixture generative models is to disentangle the data distribution into multiple components where each component represents one mode of the input distribution. Such disentanglement is useful in many tasks such as clustering (Section 6). To test the effectiveness of NWGAN in performing such disentanglement, we consider a mixture of $50,000$ images from CIFAR-10 and $100,000$ images from CelebA [12] datasets as our input distribution. All images are reshaped to be 32×32.

To highlight the importance of optimizing mixture proportion to produce disentangled generative models, we compare the performance of NWGAN with a variation of NWGAN where the mode proportion π is held fixed as $\pi_i = \frac{1}{k}$ (the uniform distribution). Sample generations produced by both models are shown in Figure 4. When π is held fixed, the model does not produce disentangled representations (in the second mode, we observe a mix of CIFAR and CelebA generative images.) However, when we optimize π, each generator produces distinct modes.

6. Adversarial Clustering

In this section, we use the proposed NW measure to formulate an adversarial clustering approach. More specifically, let the input data distribution have k underlying modes (each representing a cluster), which we intend to recover. The use of deep generative models for performing clustering has been explored in [25] (using GANs) and [13](using VAEs). Different from these, our approach makes use of the proposed NWGAN for clustering, and thus explicitly handles data with imbalanced modes.

Let $\mathbb{P}_X$ be observed empirical distribution. Let $\mathbf{G}^*$ and π^* be optimal solutions of NWGAN optimization (9). For a given point $\mathbf{x}_i \sim \mathbb{P}_X$, the clustering assignment is computed using the closest distance to a mode i.e.,

$$C(\mathbf{x}_i) = \arg \min_{1 \le j \le k} \min_{Z} \left[\|\mathbf{x}_i - \mathbf{G}_j(Z)\|^2 \right]. \qquad (10)$$

To perform an effective clustering, we require each mode $\mathbf{G}_j$ to capture one mode of the data distribution. Without enforcing any regularization and using rich generator functions, one model can capture multiple modes of the data distribution. To prevent this, we introduce a *regularization* term that maximizes the weighted average Wasserstein distances between different generated modes. That is,

$$R = \sum_{(i,j)|i>j} \pi_i \pi_j W\left(\mathbf{G}_i(Z), \mathbf{G}_j(Z)\right). \qquad (11)$$

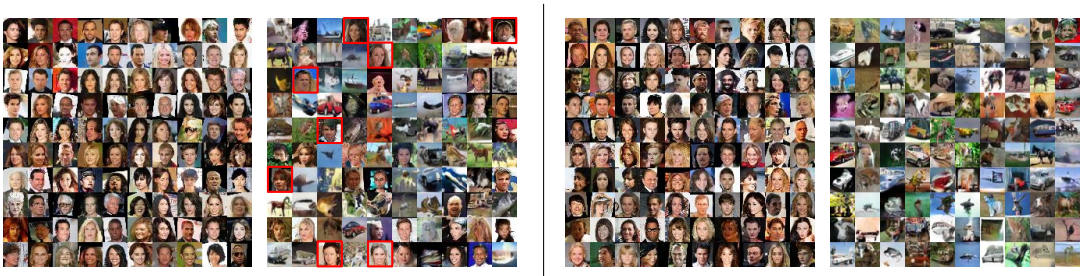

Figure 4. Sample generations of NWGAN with $k = 2$ on a mixture of CIFAR-10 and CelebA datasets for fixed and optimized π's. When π is fixed, one of the generators produces a mix of CIFAR and CelebA generative images (boxes in red highlight some of the CelebA generations in the model producing CIFAR+CelebA). However, when π is optimized, the model produces disentangled representations.

This term encourages *diversity* among generative modes. With this regularization term, the optimization objective of a *regularized* NWGAN becomes

$$\min_{\mathbf{G},\pi} W(\mathbb{P}_X, \mathbb{P}_{\mathbf{G},\pi}) - \lambda_{reg} R$$

where λ_{reg} is the regularization parameter.

We test the proposed adversarial clustering method on an imbalanced MNIST dataset with 3 digits containing $3,000$ samples of digit 2, $1,500$ samples of digit 4 and $6,000$ samples of digit 6. We compare our approach with *k-means* clustering and *Gaussian Mixture Model (GMM)* in Table 6. Cluster purity, NMI and ARI scores are used as quantitative metrics (refer to SM Section 5.3 for more details). We observe that our clustering technique is able to achieve good performance over the compared approaches.

Table 6. Clustering results on Imbalanced MNIST dataset

Method	Cluster Purity	NMI	ARI
k-means	0.82	0.49	0.43
GMM	0.75	0.28	0.33
NW	**0.98**	**0.94**	**0.97**

7. Choosing the number of modes

As discused in Section 3, choosing the number of modes (k) is crucial for computing NW measure. While this information is available for tasks such as domain adaptation, it is unknown for others like generative modeling. In this section, we experimentally validate our theoretically justified algorithm for estimating k. Consider the mixture of Gaussian dataset with $k = 9$ modes presented in Section 5.1. On this dataset, the NWGAN model (with same architecture as that used in Section 5.1) was trained with varying number of modes k. For each setting, the NW measure between the generated and real data distribution is computed and plotted in Fig 5. We observe that $k = 9$ satisfies the condition discussed in Theorem 1: optimal k^* is the smallest k for which $NW(k)$ is small, $NW(k-1) - NW(k)$ is large, and $NW(k)$ saturates after k^*.

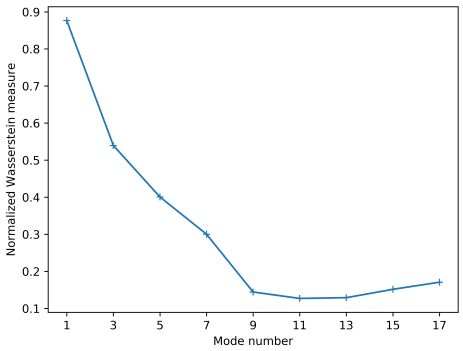

Figure 5. Choosing k: Plot of NW measure vs number of modes

8. Conclusion

In this paper, we showed that Wasserstein distance, due to its marginal constraints, can lead to undesired results when when applied on imbalanced mixture distributions. To resolve this issue, we proposed a new distance measure called the Normalized Wasserstein. The key idea is to optimize mixture proportions in the distance computation, effectively normalizing mixture imbalance. We demonstrated the usefulness of NW measure in three machine learning tasks: GANs, domain adaptation and adversarial clustering. Strong empirical results on all three problems highlight the effectiveness of the proposed distance measure.

9. Acknowledgements

Yogesh Balaji and Rama Chellappa were supported by a MURI program from the Army Research Office under the grant W911NF17-1-0304. Soheil Feizi was supported by the US National Science Foundation (NSF) under the grant CDS&E:1854532, and Capital One Services LLC.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017. 3, 6

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018. 6

[3] Qingchao Chen, Yang Liu, Zhaowen Wang, Ian Wassell, and Kevin Chetty. Re-weighted adversarial adaptation network for unsupervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 4

[4] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012. 1

[5] Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015. 6

[6] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR. 2, 4

[7] Arnab Ghosh, Viveka Kulharia, Vinay P Namboodiri, Philip HS Torr, and Puneet K Dokania. Multi-agent diverse generative adversarial networks. *CoRR, abs/1704.02906*, 6:7, 2017. 6

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 5, 6

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. *arXiv preprint arXiv:1704.00028*, 2017. 6

[10] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. MGAN: training generative adversarial nets with multiple generators. 2018. 6

[11] Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1718–1727, 2015. 6

[12] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 7

[13] Francesco Locatello, Damien Vincent, Ilya O. Tolstikhin, Gunnar Rätsch, Sylvain Gelly, and Bernhard Schölkopf. Clustering meets implicit generative models. *CoRR*, abs/1804.11130, 2018. 7

[14] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 97–105, 2015. 4

[15] Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. *CoRR*, abs/1602.04433, 2016. 4

[16] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, and Zhen Wang. Multi-class generative adversarial networks with the l2 loss function. *arXiv preprint arXiv:1611.04076*, 2016. 6

[17] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pages 271–279, 2016. 6

[18] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *CoRR*, abs/1710.06924, 2017. 5

[19] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 6

[20] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *AAAI*, pages 4058–4065. AAAI Press, 2018. 4

[21] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017. 4

[22] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008. 1

[23] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 945–954, 2017. 4

[24] Yaoliang Yu and Csaba Szepesvári. Analysis of kernel mean matching under covariate shift. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012. 4

[25] Yang Yu and Wen-Ji Zhou. Mixture of gans for clustering. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3047–3053. International Joint Conferences on Artificial Intelligence Organization, 2018. 7